

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Method and System for Centralized Network Usage
Tracking**

Inventor(s):

**Rico Mariani
Tarek Najm
Ramesh Manne
Lee Wang
Tara Prakriya
Madhan Subhas**

00704196-103100

1 **TECHNICAL FIELD**

2 This invention relates to networks and information logging, and more
3 particularly to a method and system for centralized network usage tracking.
4

5 **BACKGROUND OF THE INVENTION**

6 Computer systems throughout the world are becoming increasingly
7 connected via the Internet and the World Wide Web. The World Wide Web (also
8 referred to as simply the "Web") is a collection of documents (commonly referred
9 to as "Web pages") that users can view or otherwise render and which typically
10 include links to one or more other pages that the user can access. Web pages are
11 hosted on a web server that is accessible to client devices via the Internet and can
12 provide a wide range of information, such as company or personal information,
13 product information, interactive information allowing purchases of goods or
14 services to be made, etc.

15 Businesses and individuals often find it beneficial to be able to track the
16 manner in which users use the web (e.g., what web pages are being viewed by
17 users). Such tracking allows businesses to identify user needs and behaviors, and
18 better provide the users with the information they desire. When companies (which
19 may include multiple different divisions, subsidiaries, etc.) have a larger number
20 of web servers it is beneficial for the information logged at the individual servers
21 to be compiled into a large, centralized log. However, such compilation can prove
22 to be troublesome at best.

23 Currently it can be very difficult to compile information from a large
24 number of web servers because each web server must perform its own logging of
25 usage, those individual logs must be accessed, and the necessary information

1 According to another aspect, the logging server responds to requests with a
2 small, trivial response. In one exemplary implementation, the trivial response
3 totals only 49 bytes and is a one pixel by one pixel transparent graphic image
4 (which therefore will not affect the appearance of the rest of the web page when
5 rendered).

6 7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 The present invention is illustrated by way of example and not limitation in
9 the figures of the accompanying drawings. The same numbers are used
10 throughout the figures to reference like components and/or features.

11 Fig. 1 is a block diagram illustrating an exemplary network environment
12 such as may be used in accordance with certain embodiments of the invention.

13 Fig. 2 is a block diagram illustrating the exemplary data flow in logging
14 network usage at a centralized log server in accordance with certain embodiments
15 of the invention.

16 Fig. 3 illustrates an exemplary web page including a tracking tag in
17 accordance with certain embodiments of the invention.

18 Fig. 4 illustrates an exemplary trivial response.

19 Fig. 5 is a flowchart illustrating an exemplary process for centrally logging
20 server accesses in accordance with certain embodiments of the invention.

21 Fig. 6 illustrates an example of a suitable operating environment in which
22 at least portions of the invention may be implemented.

0 9 2 0 7 0 5 1 0 0

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

20
21
22
23
24
25

1 (e.g., in local logs 116), etc. It should be noted, however, that this local logging is
2 distinct from the centralized logging as discussed in more detail below.

3 The format of documents 114 can vary depending on the protocol(s)
4 supported by the network. By way of example, network 108 may be the Internet,
5 servers 104 web servers, and documents 114 written in HTML or XML. A
6 document 114 can optionally include references to content that is to be retrieved
7 from different sources. For example, a browser 110 accessing a document 114
8 may, upon rendering the document, obtain content from additional sources (such
9 as different documents 114 on the same server, or content on another server,
10 content cached locally at client 102, etc.). During operation, browser 110 can
11 render the content as it is received, or alternatively wait until all the content has
12 been received from the various sources prior to rendering the content.

13 One or more documents 114 include a reference to content on logging
14 server 106, which causes a browser 110 rendering the content of the document 114
15 to access logging server 106. When logging server 106 is accessed to obtain the
16 content, the access is logged by logging server 106 in log 118. Thus, accesses to
17 documents 114 on servers 104 result in additional accesses being made to logging
18 server 106, thereby allowing the accesses to servers 104 to be logged in a
19 centralized location (log 118). Although each document 114 need not include a
20 reference to content on logging server 106, documents which do not include the
21 reference are not logged by logging server 106.

22 Although the discussions herein refer to a single logging server 106,
23 alternatively multiple logging servers 106 may exist that operate together to
24 perform the centralized logging. For example, multiple logging servers may store
25 the logged data into the same log 118.

1 In one implementation, the tracking tag included in the web pages 132 is as
2 follows:

3

4 The c.msn.com is the reference to logging server 106 (that is, it is an identifier that
5 corresponds to the Internet address of logging server 106) and c.dll is the name of
6 the program on logging server 106 that performs the logging. It is to be
7 appreciated that the reference to the logging server and the name of the program
8 on the logging server that performs the logging will vary by implementation based
9 on the actual names or identifiers assigned in the specific implementation. The
10 *parameters* are logging information that is recorded by the web server 134 and
11 passed to web browser 130 along with the response 138. These parameters may
12 be implemented in different formats, such as a simple list of the information to be
13 logged, an encoded and/or encrypted version of the information to be logged, etc.

14 A wide variety of information regarding usage of the web by the user can
15 be included in the *parameters* of the tracking tag. In one exemplary
16 implementation, the following information is encoded as the parameters of the tag:
17 the subject matter of the content of the web page (e.g., soccer, sport fishing,
18 consumer audio/video electronics, etc.); an identification (e.g., address) of the web
19 server hosting the web page; what actions were being taken by the user when the
20 web page was displayed (e.g., regular viewing, filling in a form, taking a poll,
21 etc.); and what kind of advertising, if any, was being displayed to the user (e.g.,
22 the subject matter of the advertisements, specific companies/products/services
23 being advertised, etc.). All of this information is readily available to web server
24 134 and can be dynamically incorporated into the requested web pages as the
25 *parameters* of the tracking tag in a conventional manner.

1 that content for such sizes prior to rendering the web page. Thus, by placing the
2 tag outside of the formatting, the logging server 106 is not accessed by the
3 browser until after the rest of the page has been rendered (so no additional delay is
4 imposed on the rendering of the web page by accessing the logging server, nor are
5 users adversely affected if the logging server should happen to be unavailable).

6 Fig. 3 illustrates an exemplary web page including a tracking tag in
7 accordance with certain embodiments of the invention. In the illustrated example,
8 a web page 150 is written in HTML and includes a header portion 152 and a body
9 portion 154. Outside of these portions 152 and 154 is a reference portion 156 that
10 includes a tracking tag having parameters 158 (four pieces of information each
11 encoded into a two-digit number).

12 Returning to Fig. 2, logging server 106 performs its logging of data based
13 on logging request 140. In the illustrated example, trivial response 142 simply
14 serves to be a response to logging request 140. Some protocols (such as HTTP)
15 call for a response to be returned for each request – failure to return such a
16 response can result in different actions, such as another request by browser 130, an
17 error being displayed to the user that the requested content or address could not be
18 found, etc. By returning the trivial response, logging server 106 avoids
19 inefficiencies due to additional repeated requests, notifications to the user, etc.
20 Alternatively, if such inefficiencies can be tolerated (or if the protocol being used
21 does not require a response to requests) then logging server 106 need not return
22 any response.

23 The trivial response returned by logging server 106 can be any of a wide
24 variety of responses, and in one implementation is designed so that rendering of
25 the content included in the response is not perceivable by a user. By way of

1 example, the trivial response may be a single pixel that is transparent (so that
2 when displayed the pixel would not be visible to the user), or the trivial response
3 may be a single musical note with a very low or muted volume (so that when the
4 note is played it would not be heard by the user). Other types of trivial responses
5 may also be used.

6 In the illustrated example, the trivial response returned by logging server
7 106 is designed to be very small so that the performance impact on the browser of
8 having to obtain the content from logging server 106 is reduced. In one
9 implementation, trivial response 142 is only 49 bytes. By having a small trivial
10 response, the performance of logging server 106 is also improved, allowing server
11 106 to respond to a very large number of requests in a timely manner. An
12 exemplary 49-byte trivial response is illustrated in Fig. 4.

13 Fig. 5 is a flowchart illustrating an exemplary process for centrally logging
14 server accesses in accordance with certain embodiments of the invention. In the
15 illustrated example of Fig. 5, the process is described with reference to accessing a
16 web page from a web server. The process of Fig. 5 illustrates the actions taken by
17 a client web browser (portion 170), a web server (portion 172), and a logging
18 server (portion 174). These actions can be implemented in software, firmware,
19 hardware, or a combination thereof, and can be implemented in different manners
20 by the client and servers.

21 Initially, a request for a web page is sent from the client web browser to the
22 web server (act 176). The web server receives the request (act 178) and responds
23 with the requested web page including the additional tracking tag (act 180). The
24 web server also optionally logs information regarding the user (and/or information
25 about what the user was doing, viewing, etc.) locally (act 182).

1 The requested web page is received by the web browser (act 184), which in
2 turn sends out any necessary secondary requests to the appropriate servers (act
3 186). These secondary requests are to receive other content needed to display the
4 requested web page (which may be hosted on the same or different servers as the
5 requested web page). Web browser 180 renders (e.g., displays visual content,
6 plays audio content, etc.) portions of the web page as they are received (act 188),
7 and once all portions are rendered sends a tracking request to the logging server
8 (act 190). The logging server receives the tracking request (act 192) and responds
9 to the tracking request with a trivial response (act 194). The logging server also
10 logs data included in and/or corresponding to the tracking request (act 196), such
11 as information embedded in the request, cookies associated with the request, etc.
12 The web browser receives the trivial response (act 198) and renders the content of
13 the trivial response (act 200).

14 Fig. 6 illustrates an example of a suitable operating environment in which
15 at least portions of the invention may be implemented. The illustrated operating
16 environment is only one example of a suitable operating environment and is not
17 intended to suggest any limitation as to the scope of use or functionality of the
18 invention. Other well known computing systems, environments, and/or
19 configurations that may be suitable for use with the invention include, but are not
20 limited to, personal computers, server computers, hand-held or laptop devices,
21 multiprocessor systems, microprocessor-based systems, programmable consumer
22 electronics, gaming consoles, cellular telephones, public terminals or kiosks,
23 wearable computers, network PCs, minicomputers, mainframe computers,
24 distributed computing environments that include any of the above systems or
25 devices, and the like.

1 offices, enterprise-wide computer networks, intranets, and the Internet. In certain
2 embodiments of the invention, computer 242 executes an Internet Web browser
3 program (which may optionally be integrated into the operating system 270) such
4 as the "Internet Explorer" Web browser manufactured and distributed by
5 Microsoft Corporation of Redmond, Washington.

6 When used in a LAN networking environment, computer 242 is connected
7 to the local network 292 through a network interface or adapter 296. When used
8 in a WAN networking environment, computer 242 typically includes a modem 298
9 or other means for establishing communications over the wide area network 294,
10 such as the Internet. The modem 298, which may be internal or external, is
11 connected to the system bus 248 via a serial port interface 268. In a networked
12 environment, program modules depicted relative to the personal computer 242, or
13 portions thereof, may be stored in the remote memory storage device. It will be
14 appreciated that the network connections shown are exemplary and other means of
15 establishing a communications link between the computers may be used.

16 Computer 242 also includes a broadcast tuner 200. Broadcast tuner 200
17 receives broadcast signals either directly (e.g., analog or digital cable
18 transmissions fed directly into tuner 200) or via a reception device (e.g., via an
19 antenna or satellite dish).

20 Computer 242 typically includes at least some form of computer readable
21 media. Computer readable media can be any available media that can be accessed
22 by computer 242. By way of example, and not limitation, computer readable
23 media may comprise computer storage media and communication media.
24 Computer storage media includes volatile and nonvolatile, removable and non-
25 removable media implemented in any method or technology for storage of

information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other media which can be used to store the desired information and which can be accessed by computer 242. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The invention has been described in part in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks,

0070436-103100

1 although it is recognized that such programs and components reside at various
2 times in different storage components of the computer, and are executed by the
3 data processor(s) of the computer.

4 Alternatively, the invention may be implemented in hardware or a
5 combination of hardware, software, and/or firmware. For example, one or more
6 application specific integrated circuits (ASICs) could be designed or programmed
7 to carry out the invention.

8 Although the description above uses language that is specific to structural
9 features and/or methodological acts, it is to be understood that the invention
10 defined in the appended claims is not limited to the specific features or acts
11 described. Rather, the specific features and acts are disclosed as exemplary forms
12 of implementing the invention.